

AMENDMENT TO CLAIMS

Please amend claims 1, 4, 13, 15, 22, 23, 25, 30, 32, 35, 36, 37, 38, 39, 40 and 43, and cancel claim 44. All pending claims are reproduced below, including those that remain unchanged.

1. (Currently Amended) A system implemented using a computer to process XML document, comprising:

a streaming parser operable to parse an XML document to generate a stream of events discrete pieces of the XML document, wherein each event in the stream represents a portion of the document;

a matching component to perform the steps of:

accepting ~~an event~~ a discrete piece of the XML document from the stream of discrete pieces of events from the streaming parser at one time;

keeping in memory only said ~~event~~ discrete piece of the XML document from the stream of events at ~~any~~ said time;

maintaining a plurality of contexts associated with said discrete piece of the XML document;

performing a match against each of the plurality of contexts ~~that is associated with an XQuery method~~ on said ~~event~~ discrete piece of the XML document from the stream of events; and

notifying an observer when the ~~event~~ discrete piece of the XML document is a matched ~~event~~ discrete piece of the XML document, wherein when the ~~event~~ discrete piece of the XML document is not a matched ~~event~~ discrete piece of the XML document the observer is not notified;

said observer operable to listen for the matched ~~event~~ discrete piece of the XML document and passing it to a user object; and

said user object operable to handle the matched ~~event~~ discrete piece of the XML document, said matching component runs on one or more processors.

2. (Previously Presented) The system according to claim 1, wherein:
the XML document is represented in a hierarchical structure.

3. (Previously Presented) The system according to claim 2, wherein:
the hierarchical structure is a tree with each node containing a portion of the document.

4. (Currently Amended) The system according to claim 3, wherein:
the streaming parser generates the stream of ~~events~~ discrete pieces of the XML document by:
traversing the XML tree and adding visited nodes into a data structure;
processing the nodes in the data structure and generating ~~an event~~ a discrete piece of the XML document for each node; and
appending the ~~event~~ discrete piece of the XML document to the output stream.
5. (Previously Presented) The system according to claim 4, wherein:
the tree is traversed using a breath-first or depth-first search.
6. (Previously Presented) The system according to claim 4, wherein:
the data structure is a queue.
7. (Previously Presented) The system according to claim 4, wherein:
the data structure is processed using a first-in-first-out approach.
8. (Previously Presented) The system according to claim 1, wherein:
the matching component keeps only a portion of the XML document in memory at any given time.
9. (Previously Presented) The system according to claim 1, wherein:
the matching component knows the schema of the XML document and foreseeing the coming events.
10. (Previously Presented) The system according to claim 1, wherein:
the match is an expression-based match, which can be an XPath query.
11. (Previously Presented) The system according to claim 3, wherein:
the matching component keeps, clones and destroys the entirety or a portion of the sub-tree descending from a node in the tree.
12. (Previously Presented) The system according to claim 1, wherein:
the user object returns the matched event to an XML stream for use by any other component.

13. (Currently Amended) A computer-implemented method for processing XML document, comprising:

 parsing an XML document to generate a stream of ~~events~~ discrete pieces of the XML document; ~~wherein each event in the stream represents a portion of the document;~~

 accepting ~~an event~~ a discrete piece of the XML document from the stream of ~~discrete pieces of events~~ and keeping in memory only said ~~event~~ discrete piece of the XML document from the stream of ~~events~~ at one time;

maintaining a plurality of contexts associated with said discrete piece of the XML document;

 performing a match against each of the plurality of contexts that is associated with an XQuery method on said ~~event~~ discrete piece of the XML document from the stream of ~~events~~;

 notifying an observer when the ~~event~~ discrete piece of the XML document is [[a]] matched event discrete piece of the XML document, wherein when the ~~event~~ discrete piece of the XML document is not [[a]] matched event discrete piece of the XML document the observer is not notified;

 listening for the matched ~~event~~ discrete piece of the XML document and passing it to a user object; and

handling the matched event discrete piece of the XML document.

14. (Previously presented) The method according to claim 13, further comprising:

 representing the XML document in a hierarchical structure, which is a tree with each node containing a portion of the document.

15. (Currently Amended) The method according to claim 14, wherein:

 the parsing of the XML document comprises the steps of:

 traversing the XML tree and adding visited nodes into a data structure;

 processing the nodes in the data structure and generating ~~an event~~ a discrete piece of the XML document for each node; and

 appending the ~~event~~ discrete piece of the XML document to the output stream.

16. (Original) The method according to claim 15, wherein:

 the XML tree is traversed using a breath-first or depth-first search.

17. (Original) The method according to claim 15, wherein:

the data structure is processed using a first-in-first-out approach.

18. (Original) The method according to claim 13, further comprising:
keeping only a portion of the XML document in memory at any given time.
19. (Original) The method according to claim 13, further comprising:
knowing the schema of the XML document and foreseeing the coming events.
20. (Previously presented) The method according to claim 13, further comprising:
performing an expression-based match, which is an XPath query.
21. (Original) The method according to claim 14, further comprising:
keeping, cloning and destroying the entirety or a portion of the sub-tree descending from a node in the tree.
22. (Currently Amended) The method according to claim 13, further comprising:
returning the matched ~~event~~ discrete piece of the XML document to an XML stream for use by any other component.
23. (Currently Amended) A machine readable medium having instructions stored thereon that when executed by a processor to:
- parse an XML document to generate a stream of ~~events~~ discrete pieces of the XML document; ~~wherein each event in the stream represents a portion of the document;~~
- accept an ~~event~~ discrete piece of the XML document from the stream of ~~discrete pieces of events~~ and keeping keep in memory only said ~~event~~ discrete piece of the XML document from the stream of ~~events~~ at one time;
- maintain a plurality of contexts associated with said discrete piece of the XML document;
- perform a match against each of the plurality of contexts that is associated with an XQuery method on said ~~event~~ discrete piece of the XML document from the stream of ~~events~~;
- notify an observer when the ~~event~~ discrete piece of the XML document is [[a]] matched event discrete piece of the XML document, wherein when the ~~event~~ discrete piece of the XML document is not [[a]] matched event discrete piece of the XML document the observer is not notified;

listen for the matched event discrete piece of the XML document and passing it to a user object;
and

handle the matched event discrete piece of the XML document.

24. (Original) The machine readable medium of claim 23, further comprising instructions that when executed cause the system to:

represent the XML document in a hierarchical structure, which can be a tree with each node containing a portion of the document.

25. (Currently Amended) The machine readable medium of claim 24, further comprising instructions that when executed cause the system to:

parse the XML document, comprising the steps of:

traversing the XML tree and adding visited nodes into a data structure;

processing the nodes in the data structure and generating an event discrete piece of the XML document for each node; and

appending the event to the output stream.

26. (Original) The machine readable medium of claim 25, further comprising instructions that when executed cause the system to:

traverse the tree using a breath-first or depth-first search.

27. (Original) The machine readable medium of claim 25, further comprising instructions that when executed cause the system to:

process the data structure using a first-in-first-out approach.

28. (Previously presented) The machine readable medium of claim 23, further comprising instructions that when executed cause the system to:

perform an expression-based match, which is an XPath query.

29. (Original) The machine readable medium of claim 23, further comprising instructions that when executed cause the system to:

keep only a portion of the XML document in memory at any given time.

30. (Currently Amended) The machine readable medium of claim 23, further comprising instructions that when executed cause the system to:

know the schema of the XML document and foresee the coming ~~events~~ discrete pieces of the XML document.

31. (Original) The machine readable medium of claim 24, further comprising instructions that when executed cause the system to:

keep, clone and destroy the entirety or a portion of the sub-tree descending from a node in the tree.

32. (Currently Amended) The machine readable medium of claim 23, further comprising instructions that when executed cause the system to:

return the matched ~~event~~ discrete piece of the XML document to an XML stream for use by any other component.

33. (Canceled).

34. (Canceled).

35. (Currently Amended) The system according to claim 1, wherein:

said matching component can perform the step of accepting another ~~event~~ discrete piece of the XML document at said time.

36. (Currently Amended) The system according to claim 1, wherein:

said matching component can perform the step of accepting another ~~event~~ discrete piece of the XML document at a different time.

37. (Currently Amended) The method according to claim 13, further comprising:

accepting another ~~event~~ discrete piece of the XML document at said time.

38. (Currently Amended) The method according to claim 13, further comprising:

accepting another ~~event~~ discrete piece of the XML document at a different time.

39. (Currently Amended) The machine readable medium of claim 23, further comprising instructions that when executed cause the system to:

accept another ~~event~~ discrete piece of the XML document at said time.

40. (Currently Amended) The machine readable medium of claim 23, further comprising instructions that when executed cause the system to:

accept another ~~event~~ discrete piece of the XML document at a different time.

41. (Canceled).

42. (Canceled).

43. (Currently Amended) A computer-implemented method for processing XML document, comprising:
parsing an XML document to generate a stream of ~~events~~ discrete pieces of the XML document; ~~wherein each event in the stream represents a portion of the document;~~

accepting an ~~event~~ discrete piece of the XML document from the stream ~~of discrete pieces of events~~ and keeping in memory only said ~~event~~ discrete piece of the XML document from the stream ~~of events~~ at one time;

performing a match on said ~~event~~ discrete piece of the XML document from the stream ~~of events~~ and notifying an observer when the ~~event~~ discrete piece of the XML document is a matched ~~event~~ discrete piece of the XML document, wherein when the ~~event~~ discrete piece of the XML document is not a matched ~~event~~ discrete piece of the XML document the observer is not notified;

listening for the matched ~~event~~ discrete piece of the XML document and passing it to a first user object; and

returning said ~~event~~ discrete piece of the XML document to the stream ~~of events~~; and

pulling said ~~event~~ discrete piece of the XML document from the stream ~~of events~~ for the use of a subsequent user object, wherein the subsequent user object handles said discrete piece of the XML document when there is another match.

44. (Canceled)

45. (Previously Presented) The method according to claim 43, further comprising:

keeping only a portion of the XML document in memory at any given time.